

HMD (2014) 51:898–910
DOI 10.1365/s40702-014-0085-1

Sharing als Konzept, Lösung und Problem

Ein Gespräch über Informatik im technikhistorischen Wandel

David Gugerli · Andreas Meier · Carl August Zehnder · Daniela Zetti

Eingegangen: 29. August 2014 / Angenommen: 9. Oktober 2014 / Online publiziert: 25. Oktober 2014
© Springer Fachmedien Wiesbaden 2014

Zusammenfassung Als Profession und Wissenschaft war die Informatik im 20. Jahrhundert stets „eingeklemmt“ zwischen den Ansprüchen möglicher Nutzer und den Anforderungen, Möglichkeiten und Limitationen, die der Computer vorgab. Die Informatik entwickelte in diesem Spannungsfeld Sharing-Konzepte der unterschiedlichsten Art. Im folgenden Gespräch unterhalten sich zwei Informatiker (Andreas Meier und Carl August Zehnder) und zwei Technikhistoriker (David Gugerli und Daniela Zetti) über eine Entwicklung, die seit mehr als 50 Jahren anhält, die aber nicht in immer mehr Sharing kulminierte, sondern die sich stets durch lebhaften Wandel auszeichnete. Es gibt unterschiedliche Qualitäten von Sharing und es gibt und gab auch dezidierte Anti-Sharing-Entwicklungen.

Wir danken Maja Skrkic für die Transkription des Gesprächs.

Dr. D. Zetti (✉) · Prof. Dr. D. Gugerli
ETH Zürich,
Clausiusstrasse 59,
8092 Zürich, Schweiz
E-Mail: zettid@ethz.ch

Prof. Dr. D. Gugerli
E-Mail: gugerli@ethz.ch

Prof. Dr. A. Meier
Departement für Informatik, Universität Fribourg,
Boulevard de Pérolles 90,
1700 Fribourg, Schweiz
E-Mail: andreas.meier@unifr.ch

Prof. Dr. C. A. Zehnder
Chileweg 5,
5436 Würenlos, Schweiz
E-Mail: zehnder@inf.ethz.ch

1 Einleitung

Sharing ist ein Konzept, das die Informatik seit ihren Anfängen begleitet. Als der amerikanische Informatiker John McCarthy im Winter oder Frühjahr 1983 notieren sollte, seit wann er an diesem Konzept rumstudiert habe, nannte er das Jahr 1955. 1957 hätte er zum ersten Mal auch etwas Konkretes in dieser Richtung unternommen und 1959 sogar ein möglicherweise falsch datiertes, aber berühmt gewordenes Memorandum geschrieben. Erst Jahrzehnte später sei ihm klar geworden, dass ihm bei der Abfassung des Memorandums ein doppelter Fehler passiert sei. Nicht alle, die sich damals mit dem Thema beschäftigten, hätten tatsächlich dasselbe mit Sharing gemeint – manche dachten an gleichzeitig laufende Programme, McCarthy selber an die Nutzer, die sich die Rechenzeit teilen sollten. Alle aber hätten sie die für den Time Sharing-Betrieb benötigte Computerzeit unterschätzt. „I still don’t understand where all the computer time goes in time-sharing installations, and neither does anyone else.“

Time Sharing steckte also schon lange in den Köpfen der Informatiker – selbst den guten alten Batch-Prozess könnte man als Verzeitlichung des Knappheitsproblems der frühen Rechner deuten – und Time Sharing trat mit einem doppelten Paradox auf der Bühne der Informatik auf: Es war eine naheliegende Idee mit divergierenden Zielen und Zwecken, und es wollte die Allokation von Rechenzeit optimieren, um dann feststellen zu müssen, dass die Allokation selber die meiste Rechenzeit verbrauchte. Nicht weniger paradox erwiesen sich spätere Formen des Sharings. Man findet sie, wenn auch unter wechselnden Begriffen, bei relationalen Datenbanken, die seit den 1970er Jahren eine eigentliche Rekombinationskultur von Daten entwickelten, und deren Tabellen bald auch auf verschiedene Speicher und Rechner verteilt werden konnten. Oder man begegnet dem Sharing bei der Aufteilung dessen, was sich seit den 1980er Jahren im weiten Feld zwischen Servern und Clients, auf Mikrocomputern, PCs und Notebooks in Form von persönlichem Material speichern, verschieben und dadurch auch (mit-) teilen liess. Gegenwärtig taucht Sharing auf als oft zeitintensives, kaum kontrollierbares, aber äusserst mobiles Teilen von Beziehungen, Erfahrungen und Anwendungen in rechnergestützten sozialen Medien.

2 Sharing Time

Gugerli:

Wenn wir die Nutzung knapper Rechenkapazität bei den ersten kommerziellen Rechnern zum Ausgangspunkt nehmen: Wie wurden Nutzungskonflikte damals gelöst?

Zehnder:

In den 1950er Jahren wurden die wenigen Computer wie Laborgeräte benützt: Jeder Wissenschaftler plante seine Arbeit, schrieb dazu das Computerprogramm auf Papier, reservierte sich einige Stunden Rechenzeit und sass schliesslich am Rechner, beobachtete den Programmablauf und machte daran allenfalls Änderungen. In den 1960er Jahren ermöglichten Batch-Betriebssysteme die sog. Stapelverarbeitung: die automatische Abarbeitung vieler Programme hintereinander ohne jeden Programmiererein-

griff. Erst mit dem sogenannten Time Sharing-Konzept konnte der Benutzer wieder individuell am Computer arbeiten, wobei die Maschine dank dem Interrupt-Befehl (ab 1965) das Sharing organisierte. Dabei stellte das Computer-Betriebssystem mehrmals pro Sekunde mehreren Benutzern an Terminals kurze Zeitschnitze der Rechenleistung eines Grosscomputers zur Verfügung.

Zetti:

Die Entwicklung, die es ermöglicht hat, dass der Computer die Zuteilung von Rechenzeit steuerte, war also nicht die Verbreitung von kommerziellen Betriebssystemen oder die Einführung von Multiprogramming-Fähigkeiten, sondern der Interruptbefehl?

Meier:

Die Maschine ordnete. Das Betriebssystem, die koordinierende Software der Maschine, ordnete dem Benutzer einen Slot zu und dieser hatte so das Gefühl, die Maschine sei für ihn da, wobei ihm Rechenzeit nach einem Algorithmus zugeteilt wurde.

Zehnder:

Betriebssysteme entstanden anfangs schrittweise. Noch 1962 konnte ich selbständig für die „Elektronische Rechenmaschine der ETH“ (ERMETH) eine Komponente zur Aufteilung grosser Programme schreiben. Mit der Einführung des Time Sharings wurden die Betriebssysteme aber bereits erstmals zu Monstern. Dank des Interruptbefehls konnten sie zwar viele Benutzer gleichzeitig bedienen. Doch dieser Prozess war aufwendig. Damals brauchte man 80 % der Rechenleistung für die Verwaltung dieser Time Sharing-Organisation; 20 % blieben für produktives Rechnen. Eine Time Sharing-Maschine war folglich zum grössten Teil mit sich selbst beschäftigt. Für Niklaus Wirth war dieser hohe Aufwand für die Verwaltung im Time Sharing ein Grund, sich in der Entwicklung des Personal Computers zu engagieren. Wir geben jedem Nutzer einen Computer – dieses Motto war in der Frühzeit des Personal Computers auch eine Antwort auf die Probleme, die das Time Sharing mit sich brachte.

Meier:

Zu Beginn des Informatikzeitalters war der Computer ein Rechner und er hat durchgerechnet. Später kam die Möglichkeit auf, Rechenzeit aufzuteilen. Dieses Aufteilen machte wiederum weitere Anwendungen möglich. Neben wissenschaftlichem Rechnen entstanden erste kommerzielle Applikationen.

Zehnder:

Es lohnte sich nicht, ein solches Monstrum von Maschine für eine einzelne Aufgabe zu kaufen, auch nicht für die Lohnabrechnung einer Firma.

Gugerli:

Rechner waren mit andern Worten extrem teuer. Niklas Luhmann hat 1966 ein Buch über Automation in der öffentlichen Verwaltung geschrieben und darin eine schöne Beobachtung zu diesem Problem gemacht: Es gehe „ein erfrischender Denkwang

von dem glücklichen Umstand aus, dass die Maschinen so teuer sind. Ihr Preis zwingt dazu, die Organisation der Datenverarbeitung auch ausserhalb der eigentlichen Anlage in einem Maße zu rationalisieren, das ohne diesen Anstoß undurchführbar geblieben wäre.“ Weil sich einzelne Abteilungen keine eigenen Computer leisten konnten, gab es einen Zwang darüber nachzudenken, wie man den Rechner gleichzeitig für verschiedene Zwecke nutzen könnte. Dafür musste man über die Verwaltung und über die Möglichkeit zum „Sharing“ der vorhandenen Rechenkapazität nachdenken. Interessant ist aber Ihre Bemerkung, man habe nur 20% wirklich gerechnet und zu 80% habe der Computer sich selbst verwaltet.

Zehnder:

Mit meiner Dissertation habe ich die Prüfungsverwaltung der ETH des Jahres 1964/65 automatisiert. Zu dieser Zeit gab es noch kein technisches Time Sharing. Ich konnte grosse Programme einsetzen und im Stapel-Betrieb effizient arbeiten. Programmänderungen während des Rechnens waren aber nicht möglich. Ähnlich war es im „Studentenbetrieb“ der 1960er Jahre: jeder Student hatte ein paar Sekunden zur Verfügung und kam damit auch aus. Aber jeder Programmfehler musste anschliessend auf Lochkarten korrigiert und das Programm am nächsten Tag nochmals gerechnet werden.

Gugerli:

Also ein Time Sharing avant la lettre? Obwohl keine brauchbaren Time Sharing-Systeme existierten, gab es ein „Sharing“ des Computers, dessen Rechenkapazität aufgeteilt wurde.

Zehnder:

Absolut!

Meier:

Es wurden ganze Berufsgruppen beschäftigt, um die teure Maschine in geeigneter Weise zu füttern: Operateure, Lochkartenstanzer; viele Frauen. In den grossen Rechenzentren flitzten Operateure mit Rollschuhen umher, um Magnetbänder zu holen und in Lesegeräte einzuspannen.

Zehnder:

Weil die Maschine teuer war, war das Personal im Verhältnis zu ihr billig, auch in der Hochkonjunktur. 1966 war ich mit einer Delegation der ETH in den USA. Wir haben verschiedene Rechenzentren besucht und am Schluss kam mir eine wichtige Erkenntnis: es gab zwei Sorten von Direktoren an diesen Rechenzentren, Nervöse und Zurückgelehnte. Die Zurückgelehnten waren die, die noch kein Time Sharing hatten. Die Aufgeregten waren die, die es mit dem Time Sharing versuchten.

Zetti:

Worin bestand die grosse Aufregung jener Verantwortlichen, die mit computerisiertem Time Sharing arbeiteten?

Zehnder:

Die neuen Time Sharing-Systeme machten immer wieder Probleme. Darum kamen auch die Chefs der Rechenzentren unter Druck. Jene mit klassischen Stapelverarbeitungs-Systemen konnten den Benützern einen stabilen Betrieb bieten, jene mit den neuen Maschinen konnten es nicht.

Zetti:

Aber die Erwartung bei der Einführung und der Entwicklung der Time Sharing-Systeme wäre gewesen, dass man dem Operateur die Arbeit leichter macht. Oder was waren die Erwartungen an die Technisierung des Sharings?

Zehnder:

Natürlich wollte man auch den Operateur entlasten. Aber das war nicht so einfach, weil die Benutzer noch keine Bildschirme, sondern nur Fernschreiber hatten. Die Maschine selber hatte schon in den 1960er Jahren Bildschirme, auf denen man den Betriebszustand ablesen konnte.

Gugerli:

Ein tolles Bild – eine Maschine, die über sich selber mehr weiss als ihre Nutzer. Diese (Selbst-)Überwachungsform interessiert mich. Man könnte sagen, dass die Maschine dank der Zustandsanzeige auf dem Bildschirm auf raffinierte Weise sich selber und alle Vorgänge in ihr überwachen kann. Das heisst, die Maschine und ihre Operateure mussten sich um das kümmern können, was in der Maschine lief. Sie überwachten Maschine, Prozesse und Nutzer...

Meier:

„Die Maschine“ – welche Ängste hat dieses Arbeiten an der Maschine in der Gesellschaft ausgelöst? Die Maschine, die die Menschen beherrscht...

Zehnder:

Der Computer musste funktionieren und da brauchte es keine soziale Überwachung, sondern einfach Technik und Reparatur bei technischen Problemen. Ängste gab es erst ab 1973 und vor allem ab 1978; erst die neuartigen Datenbanken und Fragen des Datenschutzes haben öffentliche Diskussionen über den Einsatz von Computern ausgelöst.

Gugerli:

Aber gerade beim Time Sharing seit Mitte der 1960er Jahre musste man doch wissen, wer was wann tun darf, also brauchte man schon damals eine präzise Überwachung der Vorgänge im und am Computer.

Meier:

Ich behaupte, dass die Monstercomputer, die die Menschheit auf den Mond gebracht haben oder Therapiegespräche anbieten konnten (z. B. das Programm ELIZA 1966, Joseph Weizenbaum), sicher schon in den 1960er Jahren Ängste ausgelöst haben, auch wenn wir noch nicht sensibilisiert waren für Persönlichkeitsrecht und Datenschutz.

Zetti:

Es gab Ende der 1960er Jahre Diskussionen darüber, was diese Maschine namens Computer so neuartig machte, was sie bewirken kann und wird. Es ging darum, dass sie in Fabrikationsprozessen Überwachungsfunktionen zu übernehmen hatte. In diesem Zusammenhang wurden auch viele Fragen über Freiheit diskutiert. Wie frei ist jemand, der in einem routinierten Fabrikarbeitsablauf eingebunden ist? Zu dieser Zeit wurden Automation und Computerisierung noch mit Rationalisierung gleichgesetzt.

3 Sharing Storage and Data

Gugerli:

Wenn wir uns jetzt einer zweiten Phase zuwenden, in der weniger Betriebsabläufe im Vordergrund des Interesses standen und in der auch das Zuweisen und Teilen von Rechenkapazität weniger prominent verhandelt wurde, dann müssten wir uns über das Problem der Daten unterhalten, und zwar mit Blick auf die Medien, die in den Rechenzentren zu ihrer Speicherung eingesetzt wurden. Was das Teilungs- und Sharing-Problem angeht, so hat es sich zu Beginn der 1970er Jahre hin zum Problem des Speichers verschoben. Damit waren auch die Daten im Fokus. Wo der Speicher zur kritischen Grösse wird und von Rechenzentren mitverwaltet werden muss, entsteht ein Bewusstsein von Daten. Die sind objektivierbarer als Prozesse, man sieht sie quasi in Form von Magnetbändern und Platten. Wie wurden Daten akkumuliert? Wie wurden sie verteilbar?

Zehnder:

Es gab früh schon sehr grosse Datenbestände. Wenn man in sequentiellen Dateien, also namentlich in Magnetbändern, etwas suchte, dauerte es lange, bis man fündig wurde. Aber als die kommerzielle mit der wissenschaftlichen Datenverarbeitung in Datenbanken zusammengeführt wurde, fingen die Daten an, miteinander zu sprechen. Mit der Datenbank-Denkweise und mit dem von Edgar F. Codd 1970 erstmals beschriebenen Modell der relationalen Datenbank setzte sich eine neue Grundeckenntnis durch: das stabile und vernetzte Speichern der Daten ist die Basis aller Datenverarbeitung, der schnelle Zugriff ist nur die Ergänzung dazu.

Gugerli:

Das würde unsere Vermutung von vorhin bestätigen, dass die Daten erst im Nachhinein zum Problem wurden. Sie sind gesondert behandelt worden, zunächst von Programmierern, die wussten, was wo gelagert war, oder später mit relationalen Datenbankenmodellen und ihrer Unabhängigkeit von Daten und Nutzern. Wie hat sich dadurch die Rolle des Personals rund um die Computer herum verändert?

Meier:

Codd hat ja gesagt, dass ihn der Berechnungsprozess der Auswertung nicht interessiert. Der Anwender formuliert ein Selektionsprädikat, der Rest wird der Maschine überlassen. Ein gängiges Beispiel war folgende Abfrage: Nenne mir alle Mitarbeitenden, die mehr verdienen als ihre Vorgesetzten.

Gugerli:

Charles Bachman hat 1974 in einer öffentlichen Auseinandersetzung mit Codd vehement gefordert, dass der Programmierer weiterhin die Lufthoheit behalten müsse. Codd wollte aber den Programmierer als Gatekeeper und Navigator im Datenmeer weghaben. Einfache, verknüpfbare Tabellen, ohne Semantik und „representational clutter“ auf der einen Seite, anspruchsvolle, wenn auch technisch ignorante Benutzer auf der andern Seite. Das heisst, die Zunahme des Datenvolumens und der Speichermöglichkeiten führte dazu, dass Speicherteilungsprobleme, Zuweisungsprobleme und Abfrageprozeduren neu verhandelt und anders behandelt werden konnten.

Zehnder:

Hier kommt hinzu, dass die ersten (vor-relationalen) Datenbanksysteme Ende der 1960er Jahre auf einen schnellen Zugriff optimiert waren. Zugunsten eines schnelleren Zugriffs wurden die verrücktesten Datenstrukturen eingebaut. Das wurde zur Gefahr, wenn man eine Computeranwendung langfristig anlegen wollte. Man kann nicht langfristig Zugriffsstrukturen bereithalten. Codd hat erkannt, dass das Langfristige die Daten selber sein müssten.

Meier:

Es gab dann in den 1980er Jahren einen Quantensprung, als die relationalen Datenbanksysteme kommerziell hochkamen und leistungsfähig wurden. Beim Schweizerischen Bankverein hatte ich die Aufgabe, die relationale Datenbanktechnologie einzuführen. Ich wurde vom Management gefragt, ob ich die SQL (Structured Query Language) dem gewöhnlichen Benutzer freigebe. Information ist Macht! Ist das Management bereit, seinen Angestellten ein mächtiges Auswertungswerkzeug zur Verfügung zu stellen?

Gugerli:

Das war genau die Forderung von Codd.

Zehnder:

Nein, das war sein Angebot. Codd hat mir selber gesagt, ihm gehe es darum, dass wir damit rechnen sollen, dass die Computer beliebig schnell werden und damit auch die Zugriffe. Aber wenn wir keine sauberen Daten haben, nützt uns das nichts. Also müssen wir die Daten in der absolut einfachsten Form speichern: in Tabellen. Das sind „nackte“ Daten ohne Semantik, ohne Speicheradresse, ohne Zugriffsstruktur, aber verknüpfbar.

Meier:

Die Relationenalgebra ist die Sprache, um mit Relationen arbeiten zu können. Codds Verdienst ist, dass er als Mathematiker das symbolische Rechnen nach vorne gebracht hat. Die Behauptung und der Streit waren damals, dass man eine solche Datenbankmaschine gar nicht bauen könne. Zudem konnte man dem gelegentlichen Anwender die Relationenalgebra nicht zumuten, deshalb hat man aus der Relationenalgebra die SQL entwickelt.

Zehnder:

Das ist ein schönes Beispiel dafür, wie die Informatik eine Lücke geschlossen hat zwischen den mathematiknahen Verfahren und dem, was für Anwender wirklich brauchbar ist.

Zetti:

Das klingt ganz so, als hätte die Informatik mit Codd auch angefangen, den Kern der bis dahin selbst entworfenen Sharing-Konzepte anzutasten. Jetzt wurden technisierte und nicht-technisierte Sharing-Formen gebündelt, die eigene Leistung der letzten Jahrzehnte wurde begutachtet und es wurde Ordnung gemacht: die älteren Konzepte hatten Rechenzeit so stark in den Mittelpunkt gestellt, dass die Informatik nun eine Initiative starten musste, um Rechenzeit und -leistung als zentrale Werte wieder aus dem Fokus jener zu nehmen, die mit und an Computern arbeiteten. Zeit und Leistung waren im Prinzip beliebige Grössen.

Zehnder:

...aber nicht der Speicher, sondern die Daten selber sind die Grundeinheit.

Gugerli:

Es liess sich also argumentieren, die gelegentlichen Anwender würden bei verteilten Datenbanken Abfragen starten, durch die sie näher an die Daten rankommen – Benutzer können eigene und fremde Daten mit andern teilen. Den Programmierer, der alles überwachen soll, können sie zur Seite schieben und ignorieren. Gleichzeitig ist es aber so, dass auch die Benutzer ein gutes Stück wegrücken: vom Rechner als Rechenmaschine, vom Rechner als Speichereinheit und von den Daten als kombinierbarer Ressource. Dazwischen stehen immer ein Datenbankmanagementsystem und eine Abfragesprache. Der Rechner und die Daten werden zwar teilbarer und zugänglicher, allerdings sind sie immer besser geschützt, wie hinter einer Glasscheibe. Man kann nicht selber in die Datenstruktur einwirken, man kann sie nur nutzen. Hat dann diese neue Entwicklung in den 1970ern und frühen 1980ern von der Informatikseite und der technischen Seite her das Monster nicht noch grösser werden lassen? Obwohl man als Nutzer näher am Computer war, spürte man, dass man nicht durchkam.

Zetti:

Handelte es sich, aus der Perspektive jener Kunden formuliert, die Datenverarbeitungsaufträge erteilen, auch um eine Monopolisierung der Datenhaltung? Kunden der 1970er und 1980er Jahre gaben ihre Dateien ab und diese verschwanden dann im Computer. Der Kunde der 1950er und 1960er Jahre hatte mit seinen Lochkarten wieder heimgehen können, nachdem der Rechenjob erledigt war.

Meier:

Das kritische Element war das Verknüpfungspotenzial. Vor den relationalen Datenbanken benutzte man Karteikarten und wenn etwas kombiniert werden musste, war es mit Aufwand verbunden. Mit den relationalen Datenbanken ist ein immenses Verknüpfungspotenzial entstanden: mit einem einzigen SELECT-Statement lassen sich verschiedenste Aspekte zusammenführen. Erst diese Kombinationsmöglichkeit

schürte die Angst, namentlich wenn persönliche Daten tangiert und die Privatsphäre verletzt werden konnten.

Gugerli:

Diese Kombinationsmöglichkeit finde ich als neue Form des Sharing enorm interessant. Man teilt Datenbestände auf. Meines Wissens beginnt man in den 1980er Jahren sogar damit, Datenbanken auf verschiedene Rechner zu verteilen. Verknüpfte Rechner bilden dann die Gesamtheit der relevanten Datenbasis, die, wie Sie gesagt haben, mit einem einfachen SELECT Befehl angesteuert werden kann. Das Entscheidende bei dieser Verbindung ist, dass man die Auswahl frei treffen kann.

Meier:

Später hat sich Codd auch mit der Verteilung beschäftigt und sein Argument war, das Verteiltsein müsse den gewöhnlichen Nutzer gar nicht interessieren.

Zetti:

Es kann wohl sein, dass es den gewöhnlichen Nutzer nicht interessierte, wo die Daten verteilt gespeichert waren. Aber war die Verteilung ein Problem, mit dem die Informatik rang? Sprechen wir über die Zeit, in der die Informatik auf dieses grosse und positiv konnotierte Vokabular setzte: Information muss frei sein etc.?

Zehnder:

Dazu könnte ich mehrere Beispiele aufzählen. Das Zusammenfügen von unterschiedlichen Datenbeständen ist dabei jedoch eines der ganz grossen Entwicklungsdramen, die wir in allen Informatikbereichen hatten und heute noch haben. Ein Beispiel sind Bibliotheken, wo Hauptkataloge, Hilfskataloge und Kataloge für Spezialgebiete zusammengeführt werden mussten.

4 Sharing Personal Stuff

Gugerli:

Wenn wir auf die frühen 1980er Jahre zu sprechen kommen: Daten mussten kompatibel gemacht werden, auch wenn sie aus Bereichen stammten, die sehr oft sehr heterogen organisiert waren. Hier fand eine ganz andere Art von Sharing statt. Man hatte zwar das Gefühl, der Bau von Rechenzentren, die rasante Verbreitung von Rechnern und Datenbanken sei ein Selbstläufer, das habe nichts mit der Lebenswelt zu tun. Dennoch sind damals die Datenschutzfragen verstärkt aufgekommen. Schon vor dem Deutschen Herbst (1977) versuchte man, Polizeiarbeit rechnergestützt durchzuführen und steigerte damit die Ängste vor dieser „sauberen“ Datenwelt noch mehr. Je verteilter die Daten waren, desto unheimlicher wurden sie in der Wahrnehmung für die Nutzer. Man konnte so viel mehr mit ihnen anstellen, als ursprünglich vorstellbar und beabsichtigt gewesen war.

Zehnder:

Um 1980 herum habe ich eine Arbeitsgruppe geleitet, um die schweizerische Sozialversicherungsnummer, die „AHV-Nummer“ neu zu definieren. Die alte AHV-Nummer war „sprechend“ gewesen, da sie direkt mit Geschlecht, Geburtstag und Namen verknüpft war. 1947/48 war man bei der Schaffung der AHV-Nummer völlig unsensibel gewesen in dieser Beziehung. Es hatte sich alles am Schweizer Mann ausgerichtet. Ein Namenswechsel führte zu einer neuen AHV-Nummer und traf nur Frauen. Ich habe damals gefordert, dass die AHV-Nummer nicht sprechend, dafür lebenslang stabil sein sollte.

Meier:

Das Problem war ein Datenschutzproblem. Eine sprechende Identifikationsnummer spricht Geschichte über die Person, die dahinter steht. Die alte AHV-Nummer hatte Semantik und diese kann wechseln mit der Zeit.

Zetti:

Wenn Semantik in der Struktur vermieden wird, dann hat man auch mehr Sharing-Möglichkeiten auf der Datenebene.

Gugerli:

In den 1980er Jahren kamen Personal Computer auf die Schreibtische, zunächst zur Automation von Büroarbeit. Zugleich wurde mit ihnen auch individuelle Semantik produziert. Aber nur kurzfristig, denn sehr schnell wurden diese Geräte wieder in Netzwerke eingebunden. Damit waren sie wieder im höchsten Mass auf Standardisierung angewiesen. Der Rechner war inzwischen so sicher verpackt, dass sein Innenleben dem normalen Benutzer nicht mehr zugänglich war. Das heisst, je mehr der Computer in den lebensweltlichen Bereich kam, desto fremder wurde er, desto mehr wurde er zu einer Blackbox. Im Alltag wurde ihm zwar vertraut. Aber als Speichermedium und als Kommunikationsinstrument konnte man ihn immer weniger bestimmen. Man konnte ihn zwar nutzen, aber nur nach Vorgabe.

Meier:

Für Niklaus Wirth gehörte das Graphical User Interface und die Maus zu den wichtigsten neueren Entwicklungen für die Computerwissenschaft. Der Bildschirm war nicht mehr zeilenorientiert und mit der Maus konnte man auf Informationsobjekte zeigen. Heute müssen wir dem Computer nichts mehr eingeben, ich denke dabei zum Beispiel an Augmented Reality oder Ubiquitous Computing. Wir können mit Touchscreens und Icons arbeiten. Das Wort Computer ist praktisch schon nicht mehr da. Ein Vierjähriger versteht das Smartphone als eine Art Schaufenster. Er denkt nicht an das Gerät.

Gugerli:

Was bedeuten die sich wandelnden Rollenverteilungen rund um die „Computer“ für die zahlenmässige Entwicklung des Informatikpersonals?

Zehnder:

Hier müssen wir zwischen Informatikpersonal und Informatikanwendern unterscheiden. 1980 gab es in der Schweiz 80.000 Personen, die sich professionell als Informatiker (im breitesten Sinn) betätigten. 1990 waren es gegen 100.000, 2010 etwa 130.000.

Gugerli:

Es gibt mehr zu tun und dafür braucht es mehr Leute?

Zehnder:

Nein, denn diese Zahlen stiegen eigentlich nur schwach – von 80.000 auf 130.000 in einem Zeitraum von dreissig Jahren. Aber hinter diesen Zahlen stecken grosse Verschiebungen. 1980 waren noch Locherinnen bzw. Datenerfasserinnen dabei. In der Zwischenzeit sind gewisse Berufe vollständig verschwunden und andere Berufe haben sich entwickelt – vom Sicherheitsspezialisten bis zum Web-Designer. Nicht alle Betroffenen haben solche Umstrukturierungen gut überstanden. Ein ganz anderes Thema sind die professionellen Informatikanwender in allen Berufen, vom Bankangestellten bis zum Automechaniker. Ihre Zahl ist in der Schweiz von 1980 bis etwa 2005 explosionsartig angestiegen. 1980 gab es 300.000 Anwender, 1990 etwa 800.000 und 2000 ungefähr 2,3 Mio. Dieser dramatische Zuwachs ist inzwischen weitgehend beendet. Wir sind heute bei weit über drei Millionen Informatikanwendern (von insgesamt etwa 4,5 Mio. Berufstätigen) angekommen.

Meier:

Einige Berufsbilder verändern sich aufgrund von Social Software. Früher musste man, um eine Webpage programmieren zu können, Informatiker sein. Heute kann man eine Website mit Open Source Produkten selber machen. Wir sind zwar kaum gewachsen, was die professionellen Informatiker betrifft. Hingegen hat die IT Literacy zugenommen. Es gibt Benutzer, die sich durchmausern mit ihren Fähigkeiten und solche, die entsprechende Instrumente professionell beherrschen.

5 Sharing Relationships

Gugerli:

Wir haben vorhin davon gesprochen, dass Programmierer seit den 1970er Jahren ihre Macht an den so genannten gelegentlichen Anwender, den Casual User, abgegeben haben. Was passiert jetzt bei dieser moderaten Zunahme der Spezialisten im IT-Bereich und des Mächtigwerdens der Casual User? Sind die Spezialisten noch mehr Generalisten geworden oder gibt es jetzt viele ausdifferenzierte Informatikspezialisten, die Beziehungen zwischen Rechner, Menschen und Daten organisieren?

Zehnder:

Das aktuelle Problem ist Folgendes: Heute sind die professionell in der Informatik Tätigen noch immer zu 70% Quereinsteiger. Auch wenn man 25 Jahre dabei ist,

aber keine echte Informatikausbildung hat, kann man den Entwicklungen, die wir vorher gesehen haben, nicht genügend folgen. Wenn wir von Informatikern hören, die arbeitslos sind, sprechen wir von Leuten, die sich nicht weitergebildet haben, die keine seriöse Grundausbildung in der Informatik haben und die sich nicht als Informatiker bezeichnen sollten.

Zetti:

Wenn wir sagen, dass momentan viele Abkoppelungen zu beobachten sind und dass der Computer am Schluss im iPad oder im Smartphone aufgeht: Müsste das nicht heissen, dass die Programmierer irgendwie und irgendwo in der Software auf diese Nutzer treffen? Hinterlassen diese Smartphone-Nutzer Spuren oder Materialien zuhänden der Informatik?

Meier:

Der moderne Benutzer geht auf den Informatiker zu oder er legt sich dessen Handwerkszeug zu, weil er etwas bauen will, Stichwort Apps.

Gugerli:

Da kommen ja noch ganz neue Formen des Sharings auf. Apps haben unglaublich tiefe Preise. Diese beruhen auf der minimalen Partizipation des Einzelnen an den Entwicklungskosten und auf den skalenökonomischen Effekten eines Massenmarkts. Nur so funktioniert diese Sharing-Ökonomie der Apps. Daneben gibt es die Sharing-Ökonomie in der Cloud. Speichermedien und Rechenkapazität werden von ganz unterschiedlichen Nutzergruppen geteilt. Das Sharing steckt nun in Betriebsabläufen und funktioniert als softwareunterstützte Delegation von Speichern und Rechnern. Das führt auch zu einem verteilten Arbeiten an Projekten – rund um die Uhr und rund um die Welt. Das Sharing kriegt damit nochmals ein ganz andere Dimension. Ist das eine brauchbare Deutung?

Zehnder:

Ich bin bei Sozialen Netzwerken und bei Clouds ein Verweigerer und ich weiss, dass ich eine Ausnahme bin.

Gugerli:

Inwiefern ein Verweigerer?

Zehnder:

In dem ich bei diesen Angeboten des Sharings nicht mitmache. Ich bin nicht in der Cloud. Ich speichere alle meine Daten noch selber auf meinem Computer.

Gugerli:

Weiss ich, wo meine Daten gespeichert sind? Ich glaube eher, dass es bei der Inbetriebnahme von Geräten heute schwierig geworden ist sicherzustellen, dass man nicht auf der Cloud ist.

Meier:

Ich bin durch meine Arbeit, wenn ich wissenschaftliche Dokumente austausche mit Kollegen, mit denen ich weltweit zusammenarbeite, gezwungen, solche technischen Sharing-Angebote wahrzunehmen. Dem Beitritt zu einem Sozialen Netzwerk habe ich mich bislang verweigert, obwohl ich mehrmals pro Woche Einladungen erhalte.

Gugerli:

Wir sind an einem ähnlichen Punkt, wie damals Charles Bachman, der das Gefühl hatte, die Welt breche zusammen, wenn hierarchische Strukturen oder Netzwerkdatenbankstrukturen fehlten, und Verteiltes, Relationales an ihre Stelle tritt, denn dann ist da ja keine festgefügte Ordnung mehr. Wir sehen, dass die älteren Privatanutzer versuchen, sich von den Clouds wegzuhalten, aber draussen in der Welt will man sie in der Welt der Cloud.

Zehnder:

Das stimmt aber für viele seriöse Leute und namentlich für Firmen überhaupt nicht. Die machen das Gegenteil. Die sind verantwortlich für ihre Daten, etwa die Banken und die Fluggesellschaften. Sie betreiben völlig separate Systeme. Sauber abgegrenzte Systeme sind in vielen Köpfen nicht mehr vorhanden, obwohl sie in der Realität existieren. Und ich selbst erlaube mir, meinen persönlichen Teil gesondert zu halten.

Gugerli:

In Bezug auf die Sharing-Kultur ist das ein enorm interessantes Statement. Ich habe bislang immer gedacht, dass es persönliche und unterschiedlich motivierte Verweigerungshaltungen gibt. Was Du jetzt gesagt hast, heisst, dass die Verweigerungshaltung zur operativen Voraussetzung von Unternehmungen wird. Es ist keine private Entscheidung, nicht vernetzt zu sein.

Meier:

Sie sind vernetzt, aber abgeschottet vom Rest.

Gugerli:

Auf dem Höhepunkt der Sharing-Kultur beginnt man, mit Firewalls zu arbeiten.

Zehnder:

Nicht nur mit Firewalls, sondern effektiv getrennt, für eine absolute Abschottung.